网络服务功能链技术研究与建模 *

李源灏，毛玉明，韦云凯

(电子科技大学 信息与通信工程学院, 成都 610000)

摘　要：伴随着数据业务量的急剧增长，传统的静态网络服务模型日渐不能满足应用的需求，其构架中网络服务与专用硬件设备之间一一对应的紧耦合关系造成了网络资源不能共享、新业务难以融合等弊端。当网络规模扩大时，传统的管理与部署方式需要投入更多的成本来部署新业务。随着网络虚拟化的兴起与软件定义网络研究的推进，网络服务功能链这一概念应运而生。网络服务功能链由一个或多个虚拟网络功能组成，旨在降低网络建设与运维成本、提高网络资源利用率、提升网络与业务部署速度。针对当前服务功能链体系的研究及建模与应用进行阐述。并对业内主流的服务功能链优化模型进行抽象建模。同时针对时延优化问题进行仿真。以启发式贪婪算法为代表，通过不同的启发值，从算法时间复杂度与优化效果的角度将服务功能链中启发式算法与穷举法进行对比。在网格状网络的环境下，启发式算法能取得更快的响应速度，并且能取得和穷举法相当的优化效果。因此，出在大规模工业互联网中部署服务功能链过程中启发式算法能有更快的响应速度且能达到或者接近最优解的结论。最后，就当前建模与部署存在的问题以及服务功能链未来的研究方向进行论述。

关键词：服务链；网络功能虚拟化 ；软件定义网络

中图分类号：TP393　　doi: 10.3969/j.issn.1001-3695.2018.05.0347

Survey and Modeling on Network Service Function Chain Technology

Li Yuanhan, Mao Yuming, Wei Yunkai

(*School of Communication & Information Engineering*, *University of Electronic Science & Technology*, *Chengdu* 610054, *China*)

Abstract: With the rapid growth of traffic data, the traditional static network service model is increasingly unable to meet the needs of the applications. In traditional network architecture, the relationship between service and dedicated hardware devices is tightly coupled, which results in the drawbacks of network resources cannot share and new network business is difficult to integrate. As the network continues to expand, traditional management and deployment requires more cost to deploy new services. With the rising of Network Virtualization and the development of Software Defined Network, the concept of service function chain has come into being. A service function chain is composed of one or more virtual network functions to provide a complete end-to-end service for the network. The service function chain aims to reduce the cost of network construction and operation and improve the utilization of network resources and improve the speed of network and service deployment. In this paper, the current service function chain system research and its application in industry are elaborated The development and application of service function chain are summarized. This paper will put forward a heuristic greedy algorithm with two different heuristic values. And comparing the exhaustive method in the time complexity and results of delay optimization. In the result, we can figure out that the heuristic greedy algorithm is more suitable for industrial network. Finally, this paper will discuss the modeling and drawback of service function chain present in present.

Key words: Service Function Chain; NFV; SDN

0　引言

随着互联网的高速发展，截至 2017 年 12 月，我国网民规模达 7.72 亿，全年共计新增网民 4 074 万人。若仍采用传统网络架构，则需要部署与网络功能紧耦合的专用设备来扩展网络规模。该方法不仅成本高且部署周期长，因此禁锢了网络规模的快速发展与服务的稳定性[1]为了解决网络发展速度与专属硬件部署周期不匹配的问题，研究者们提出了基于网络虚拟化的 NFV 技术。该技术利用 IT 虚拟化技术来保证网络功能的快速部署与业务处理，使用通用型硬件以及中间件来承载业务以及

收稿日期：2018-05-07；**修回日期**：2018-06-20　　**基金项目**：国家教育部—中国移动联合科研基金资助项目（MCM20160304）

作者简介：李源灏（1994-），男，福建福州人，硕士，主要研究方向为服务功能链（201621010502@std.uestc.edu.cn）；毛玉明（1956-），男，教授，主要研究方向为计算机通信网技术；韦云凯（1979-），男，副教授，主要研究方向为无线移动网络、物联网．

执行流量数据处理。通过该技术能有效减少专属网元设备的数量，从而降低部署网络成本，提高网络的灵活性与服务的稳定性[2]。

随着 NFV 技术的推进，不同的 VNF 部署方式会带来不同的效果，若缺少有效的网络服务编排方案，NFV 技术也可能造成网络拥塞以及大量网络资源的浪费。在这样的背景下，传统的网络管理协议如 SNMP、IPMI、SIM-S 等也伴随着新技术的发展被广泛应用到 NFV 网络中。在 2015 年前后，诞生了基于 NFV 技术的服务功能链（service function chain，SFC）技术，该技术旨在解决 NFV 网络中服务部署的优化问题并提供有效的端到端服务。在端到端网络中发送数据时，数据包需要经过各种业务节点才能完成数据报文的交付。服务功能链的作用就是按照用户的需求以及业务的逻辑动态地分配相应业务节点来承载业务链，最终引导数据包到达目的端。当前服务功能链的研究与使用善处于起步阶段，本文主要围绕网络虚拟化中服务功能链技术的研究与建模展开论述。通过引入贪婪算法作为服务功能链快速编排算法的代表，与全局最优穷举法在时间复杂度与优化效果两方面进行比较与分析。

1　服务功能链研究背景与现状

1.1　服务功能链研究背景

服务功能链技术这一概念是由欧洲电信标准化协会（ETSI）提出，旨在将网络中的虚拟服务功能进行合理的编排从而构建起完整的端到端服务。现今，服务功能链技术已引起业界的广泛关注。不仅因为网络服务功能链技术能够更好地满足用户和管理的需求，更因其有着减少资本支出和运营成本的巨大潜力。其通过动态部署虚拟网络功能（virtual network functions，VNFs）到通用型硬件，避免了因为新业务拓展造成垂直烟囱群不能共享资源，融合业务等问题[3]。

服务功能链旨在解决与服务功能部署的相关问题。比如防火墙需要放置在网络边缘或者重新设计网络拓扑结构来引导流量通过防火墙。这种网络强依赖性对服务的交付产生了限制，从而降低网络的灵活性，难以高效地利用网络服务资源。传统的网络功能部署方式由于需要更多可能具有相互依赖性的服务与功能，因此每加入一个新的服务与功能都可能需要更改整个网络功能的先后顺序，从而导致更复杂的网络和更多的设备配置。这样不仅限制了网络的规模、容量，更产生较多的冗余链路。同时，由于拓扑耦合限制了服务功能的放置和选择[4]。因此，需要考虑服务放置的地理位置和服务功能的选择。又如，使用服务器负载均衡器作为默认网关的 Web 服务器。当 Web 服务响应非负载均衡的流量时，来自服务器的所有流量都必须遍历负载均衡器，迫使网络创建复杂的路由方案或提供额外的接口以满足服务的需求。而服务功能链技术可以完美地解决上述问题。

1.2　服务功能链研究现状

与传统的网络相比，服务链的报文封装允许服务路径选择，并在上下文中共享元数据。然而，SFC 的封装不是一个用于传输的封装，因此数据包需要依赖外部的物理网络，具体如图 1 所示。

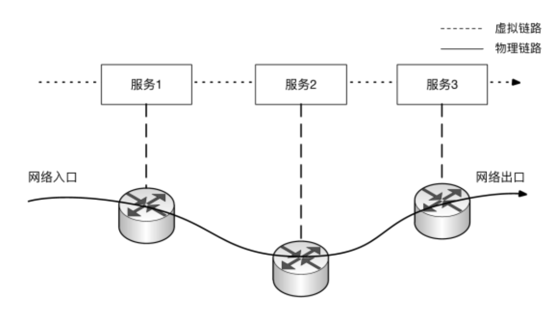


图 1　NFV 服务链示意图

在当前服务功能链的研究中，其优化与分配问题可以分为三个部分：服务功能链组成问题、服务功能链映射问题和服务功能链编排问题。由于服务功能链映射的方式具有多样性，因此，在同一个拓扑下可能存在多个编排方式可以实现相同的功能。同时，由于一些服务的特殊性（如防火墙必须放置于网络的边缘），因此服务功能的部署存在局限性。服务功能链的编排分为离线静态部署方式和在线动态部署方式。其中在线部署方式或因为终端的移动而对编排算法的时间复杂度提出更高的要求，如移动设备跨基站、车载高速设备等。如何降低算法的时间复杂度是这类问题讨论的重点[4]。以上分类方式可以归纳如图 2 所示。

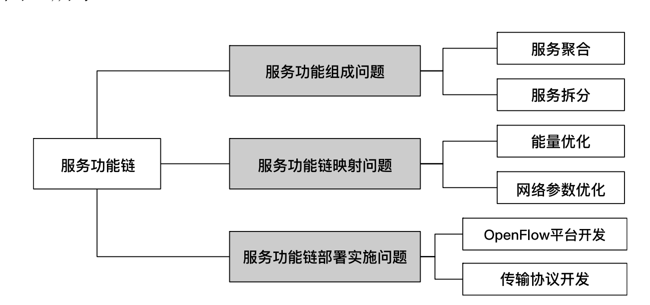


图 2　服务功能链主要研究范畴

迄今，服务功能组成这一问题研究较少，当前只有 Paderborn 大学的 Mehraghdam 教授等人研究。其在文献[5]中提出一种上下文无关服务功能链组成策略。目前绝大多数研究都假设服务功能映射到物理拓扑之前服务功能链就已组成，即直接研究服务功能链映射问题。如文献[6]就通过整数线性规划算法，并引入贪婪算法来减少网络部署的 OPEX。文献[7]为了降低当前 NFV 网络中存在的能耗与开销问题，就提出了博弈论模型来实现 SDN 网络中服务功能的开关，进而降低能耗。文献[8，9]提出了一系列启发式算法，通过动态分配 Middleboxes 来实现网络时延的降低以及带宽消耗的减少 ；文献[10]通过整数线性规划的方法，优化了服务提供的顺序从而降低了网络整体的带宽消耗。文献[11]提出了最优化求解的面向 QoS 的动态 SFC 模型。服务功能链部署与实施问题多研究与探讨服务功能链的报文与帧结构开发，如文献[12]开发并讨论了云环境下通过 OpenFlow 部署的服务功能链的报文格式。文献[13]通过最短路算法进行服务功能链映射并通过启发式算法对服务进行分级。

但是，至今仍未有文献从部署优化算法的复杂度上对服务功能链映射问题进行研究。

2　抽象建模与算法

SFC 映射与优化设计中的一个核心问题[14]是对 NFV 资源（NFV-RA）进行最优配置，以便提供高效的网络服务。其思路是如何组合出一条最佳的 VNFs 链，并将其分配给底层网络设备，为应用程序提供端到端的质量保证。考虑到虚拟网络之间的依赖关系，映射与分配具有一定挑战性。在网络设备中分配所需的网络服务，通常称为网络功能虚拟化-资源分配问题。由于多条链可以实现相同的网络服务（network service，NS），所以 VNFs 的顺序通常是灵活的；同时，某些 VNFs 必须放在特定的顺序中，例如加密的网络流首先要被解密，然后才能进一步处理。因此，当前考虑的重点就是如何建立一个自动的，动态的 SFC 体系结构。于是 IETF 就对 SFC 体系进行了抽象和概括，并认为 SFC 体系主要由：服务分类器、服务功能转发器以及服务功能等几个主要逻辑模块构成，并由 SFC 封装与代理来串联各个模块[15]。研究 SFC 映射问题的关键就是如何有机的将服务功能逻辑嵌入到底层物理网络中，并且在能够保证新业务策略请求出现的时候，提供及时的响应。

2.1　数学模型

2.1.1 物理拓扑建模

本文将总结服务功能链一般的建模方式并对其进行描述与分析。在服务功能链建模过程中，一般将物理网络表示为无向图 $G=(V,E)$，其中 V 代表物理网络拓扑上的节点，E 代表连接这些节点的物理链路。用 R 表示物理节点为虚拟网络功能提供的资源，这些资源包括节点的计算能力以及存储资源等。假设两个物理节点 u，v 本文用（u，v）表示他们间的物理连接，同时用集合 L 表示物理拓扑的物理连接集合，对任意的节点 u，v 有 $(u, v)\in L$。对物理链路（u，v）本文分别用 $B_{u,v}$，和 $D_{u,v}$ 表示他们的带宽与时延。

2.1.2 服务链虚拟网络功能

在服务功能链中可能会引入诸如防火墙（FW），IDS，IPS，Proxy 等诸多网络服务，并自由地组合成一条或多条服务功能链。本文用集合 S 表示这些服务功能种类的集合。同时每种服务功能 $s\in S$ 都有各自的部署与处理花销 D_p

本文将网络中的每一组服务功能链所需要的服务与网络的入口以及出口用一组有序的组合来表示 $sfc=\{i_{sfc},a_{sfc},b_{sfc},\ldots,e_{sfc}\}$，其中 i 和 e 分别表示该条服务功能链进入网络的入口与出口。例如建立 VPN 服务，本文用 $sfc_{VPN}=\{i_{sfc},a_{sfc},b_{sfc},c_{sfc},e_{sfc}\}$ 来表示。其中 a_{sfc},b_{sfc},c_{sfc} 分别代表虚拟网络功能提供的防火墙，IDS 以及 Proxy 这三种网络服务，并且这三种服务的先后次序存在依赖关系。因此服务虚拟功能的部署可以用有向图 $G^{SFC}=(V^{SFC},E^{SFC})$ 来表示其中 V^{SFC} 表示部署有服务功能的节点，E^{SFC} 表示他们之间有序的连接。

为了符合一般的网络场景，本文假设当前网络中已经有一定数量的服务功能链已经部署在上面，从零时刻开始又引入任意 n 条服务功能链。本文将从网络时延入手，考虑这种环境下网络的整体优化与服务部署。

2.2　优化模型与约束

当前，在业界服务功能链中虚拟功能编排的优化是一个十分复杂的问题，因为 VNF 的编排不但要考虑网络负载能力还要考虑服务的先后次序;服务功能链的部署更被研究者看做是 NP 困难问题[16]。下面本文将采用整数线性规划的约束与优化目标进行建模，并采用贪婪算法求解与分析该问题。

在在线实时部署服务链的场景下，针对动态变化的网络中的一个时刻，建模前先重建当前物理拓扑连接，将物理资源 R 已经耗尽的物理节点移除网络，并构建新的网络拓扑图，如图 3 所示。

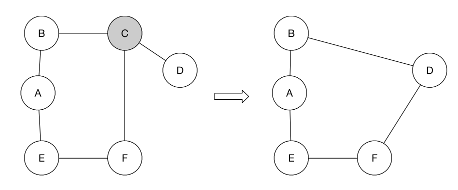


图 3　物理拓扑转换示意图

之后，引入网络流守恒约束来进一步简化物理链路拓扑间的关系。用集合 M 表示在新拓扑环境下服务功能链提供的服务集合，有任意类型服务 $m\in M$。定义一个决策变量 x_{sv}^{t} 来表述当前物理节点的承载能力变量，若物理节点 $e\in E$ 可以承载类型 m 的服务则令 $x_{sv}^{t}=1$；否则令 $x_{sv}^{t}=0$。

同时，使用二元变量 y_n 来表示激活的物理节点，若该物理节点部署有服务功能则视为激活并令 $y_n=1$，否则令 $y_n=0$；至此本文可以用下面的关系式表示 y_n 与 x_{sv}^{t} 之间的联系：

$$y_n=1?\text{当}\sum_{t\in T}x_{sv}^{t}>0$$

在此，为了确保每个节点的负载的服务不会超过其负载能力，则有约束：

$$\sum_{t\in T}x_{sv}^{t}\times D_{p,s}\le R_v\quad \forall s\in S\mid y_n=1$$

接着，本文引入网络流守恒约束，来确保每个物理网络节点流入的流量与流出的流量一致，并最终保证网络入口流入的流量与网络出口流出的流量一致。同时使用二元变量 $w_{u,v}^{ts1,s2}$ 表示服务 s_1,s_2 间的业务流是否经过物理链路（u，v）$\in L$，若经过则令 $w_{u,v}^{ts1,s2}=1$；否则 $w_{u,v}^{ts1,s2}=0$。并使用 $z_{s,v}$ 来表示服务 s 是否部署在物理节点 v 上，若部署则置 $z_{s1,v1}=1$，否则置 0。至此，可以用流守恒约束做如下表述：

$$\sum\left(w_{u,v}^{ts1,s2}-w_{v,u}^{ts1,s2}\right)=z_{s1,v1}-z_{s2,v2}\quad \forall t\in T,n_1,n_2\in N;u,v\in E`$$

最后引入如下约束来确保服务功能链是一条连续的通路，有

$$\sum\left(w_{u,v}^{ts1,s2}+w_{v,u}^{ts1,s2}\right)\ge 0?\forall t\in T,s_1,s_2\in S;$$

在服务功能链部署优化中，主要通过研究不同部署方案对网络 QoS，SLA 等的影响。本文针对时延优化与穷举法进行对比。优化目标如下，假设链路（u，v）间的时延为 $d_{(u,v)}$，链路总体时延为 D_k 则有目标：

$$\min D_k = d_{(i,v)} + \sum_{\forall s \in S} \sum_{\forall (u,v) \in L} h_{s,v} \times d_{(u,v)} + d_{(u,e)}$$

2.3 启发式-贪婪算法

根据之前的研究[17,18]可知，在数量庞大的请求来临或复杂的网络环境下其组合方式趋于无穷，因此引入启发式算法可以进一步提高收敛速度。贪心算法的思想是基于迭代的方式确定每个虚拟服务的部署位置。对于每一步迭代，不但要考虑虚拟服务在服务功能链中上下依存的关系，更要考虑物理拓扑间的联系。并在部署这些功能的时候尽量网络整体时延最小，即令服务流从最短路之间流过以获得最小的时延。其算法步骤如下：

输入：虚拟服务功能集合 M；物理拓扑 G=（V，E）；服务（链）请求集合 Se；服务的等级评分 Score=0。

输出：服务类型在物理网络上放置的位置。

1. 统计 M 中各个服务类型在请求中所占的比例，并按降序排列。

2. 针对每个类型的服务迭代置入激活的物理节点

初始化，令 $Score_{min} = inf$

$for \quad e_i \in E$

计算 $Score_i$

$if \quad Score_i < Score_{min}$

$$Score_i = Score_{min}$$

3.计算每个节点剩余资源；重新更新物理拓扑；输出服务类型 m_j 部署的位置 e_i；

算法中 $Score_i$ 的计算方式一般根据具体环境与场景甚至需要多次实验对比后才能确定。本文这里采用两组 $Score_i$ 为参照，其中 $Score_i 1$ 来自文献[19]，$Score_i 2$ 是本文根据 $Score_i 1$ 在 Mesh 网络中的劣势，并根据多次实验得出的结果。$Score_i 1$，$Score_i 2$ 的表达如下：

$$Score_i 1 = \sum_{e_i \in E} \sum_{m_j \in M} d_{i \to m} + d_{m \to m_j} + d_{m \to e}$$

其中：$d_{i \to m}$ 为部署该服务后，该物理节点到入口节点的距离；$d_{m \to m_j}$ 为部署该服务后，该物理节点到已部署节点的距离；$d_{m \to e}$ 为部署该服务后，该物理节点到出口节点的距离。

$$Score_i 2 = Score_i 1 + \frac{1}{order_{max}}$$

$Score_i 2$ 中的 $order_{max}$ 即包含 m_i 的服务链中，m_i 的位置次序。以上算法主要包括两个步骤：a）按照请求中出现的服务多寡以及重要性对每个服务 $m_i \in M$ 进行打分；b）讨论如何进行迭代，每次迭代都在保证资源约束的条件下，使时延最小。

3 分析与比较

针对诸如移动社交网络，工业大数据智能制造网络等实时在线且时延敏感的场景，以下将针对服务功能链服务编排算法的时间复杂度以及时延进行仿真与评估。本文先从数学的角度对穷举法与启发式算法在时间复杂度层面进行比较。服务功能链的优化问题可以看成是“组合优化问题”，任何一个组合优化问题不仅可以使用穷举法得出最优解[20]。而且，也可以使用智能算法迫近最优解。

为了比较这两种算法的时间复杂度，本文将图 G 的顶点数量 n，网络请求的服务策略数量 m，服务类型数量 k 三个维度作为输入长度进行分析比较。为了得到全局最优解，穷举法的思想是先将服务策略进行穷举，而后排序。首先服务策略有 $m!$ 种排序方式。而后根据排序先后顺序将每条策略嵌入到底层网络中，由于服务策略中的每个服务在每一个节点都需要比较节点的容纳能力与时延，因此需要进行 An 次运算，其中 A 是策略组每条策略中服务数量的集合之和。在每次嵌入服务策略的过程中还需要记录服务种类数量的减少，因此每条排序需要进行 Ank 次运算。将所有排序结果输出后利用冒泡法排序求最优解可以得到穷举法的计算次数 $T1$ 如下：

$$T1 = m!Ank + \frac{m!Ank(m!Ank - 1)}{2}$$

n 维度的时间复杂度为 $O(n^2)$；m 维度的时间复杂度为 $O(m!^2)$；k 纬度的时间复杂度为 $O(k^2)$ 可见穷举法是在策略数量维度上是指数时间算法，在物理节点数量与服务种类上是多项式时间算法。

同理，贪婪算法如上所示只需要进行 mk 次比较，再针对每个服务进行排布，其计算次数 $T2$ 如下：

$$T2 = \frac{mk(mk - 1)}{2} + nk$$

n 维度的时间复杂度为 $O(n)$；m 维度的时间复杂度为 $O(m^2)$；k 纬度的时间复杂度为 $O(k^2)$ 可见启发式-贪婪算法在三个维度上都是多项式时间算法，在时间复杂度上明显优于穷举法。现考虑在一个网络中，服务数量不变，即 m,k 为常数的时候，随着网络节点数 n 的增长，可以得到计算时间消耗如图 4 所示。当网络拓扑固定，请求策略增加的时候可以得到计算时间如图 5 所示。

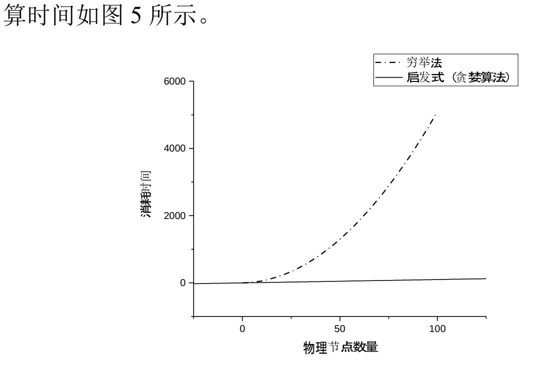


图 4 n 维度消耗时间比较

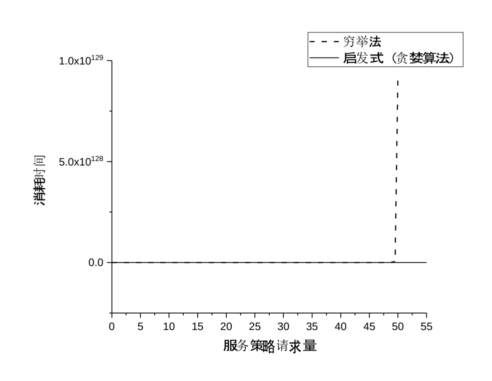


图 5　*m* 维度消耗时间比较

从图 4、5 中可以知道，随着物理节点数量 *n* 以及服务请求数量 *k* 的增长，穷举法所消耗的计算部署位置的时间将呈指数式增长。假设 NFV 网络中负责编排服务功能链的服务器计算速度为每秒 1 000 亿次，那么在 100 个物理节点，服务链请求数量为 30 条的情况下。穷举法虽然可以取得最优化的服务编排方式，但是服务器的计算时间约为 5×10^{6} s，远大于启发式算法的编排时间。因此，在大规模工业互联网中若采用穷举编排算法，一旦加入新的智能生产设备或工业机床，网络中断服务的时间较长。

智能算法虽然编排时间短，但往往容易陷入局部最优解。为了和最优算法比较该问题，这里将时延标示为网络整体的跳数，并将当前物理层剩余资源表示为 16 个节点的 Mesh 状网络作为底层物理网络的拓扑，并赋予每个物理节点 10 个单位的计算能力，每条链路通信的时候产生 1 个单位的时延代价。

以下本文将引入业务请求交叉与业务请求不交叉这两种业务请求情况分析与讨论服务功能链模型与需要解决的问题。在下面的实验中，本文将讨论启发式算法与穷举法的优势与问题。首先，本文对业务请求不产生冲突的服务功能链集合进行 MATLAB 仿真[21]。在此，本文引入 30 条服务请求，并假设其中存在 3 组不同类型服务请求，分别为{SF1 SF2 SF3};{SF4 SF5 SF6};{SF7 SF8 SF9}，每组业务的请求条数为 10 条。其次，本文针对业务请求产生交叉的服务功能链集合进行仿真。在此，同样引入 30 条业务请求，分别为{SF1 SF2 SF3};{SF4 SF2 SF5};{SF6 SF2 SF7}，同样每组请求条数为 10 条。仿真结果如图 6 所示。

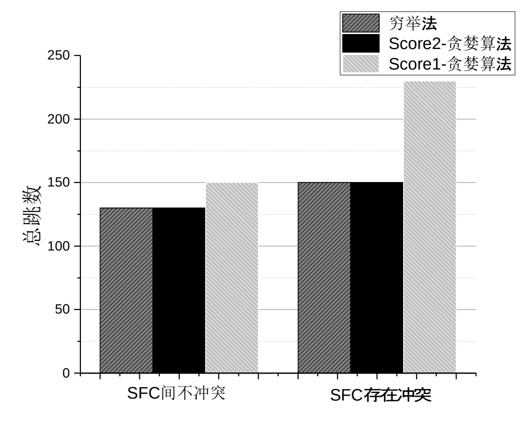


图 6　两种算法端到端跳数比较

从图 6 中可以知道，在不同的 score 取值条件下，仿真得到的最优解是不同的。这也说明了服务功能链优化部署策略当前存在的一个问题，在大规模部署服务的时候，穷举法时间复杂度高，往往不能满足工业中服务功能链动态部署的需求；在 NP 问题难以获得最优解的情况下，启

发式算法的特点是收敛速度快。但是，某些时候启发式算法可能会因为启发值的影响而偏离最优解陷入局部解。因此相比于最优解算法，在工业互联网环境下，选择时间收敛速度快的启发式算法进行服务功能链动态编排能避免生产中断，从而获得更大的经济效益。

4　结束语

作为一种新兴的服务功能管理模式，当前服务功能链也面临着诸多的挑战。首先是端到端服务的可见性（limited end-to-end service visibility）。网络故障的排除是一个复杂的过程，涉及网络与服务部分的专业知识。当服务功能链跨越多个数据中心或者管理边界的时候，这种问题会变得更加严峻。所以，就需要在网络部署上考虑到这方面问题。

其次，在部署服务功能链的时候，服务功能链的拓扑依赖关系直接决定了网络的复杂性，如何在满足服务功能的条件下尽量减少不利于网络部署的依赖关系也是当前面临的挑战。如在服务功能链中更改服务功能顺序之类的简单操作，就需要更改逻辑或物理拓扑。然而，在一些对即时性要求较高的网络如工业互联网中这一类的问题就显得尤为突出。当需要在生产环境中部署服务时，一旦出现错误配置则会导致设备停机，给生产部门带来巨大损失。此外，这些拓扑的变化速度还不够动态，在一些特殊环境中依然无法满足应用的需要。

最后，由于当前服务功能链缺乏流量选择标准。当前服务功能链的流量选择是粗泛的，这就意味着某些特定网段上的业务流量不管是否需要搜有的服务都要遍历整个服务功能。因此如何在开销最优的条件下灵活地部署服务功能链也是当今的一大挑战。随着 NFV 技术的发展，相信会有越来越多的服务与功能将移植到虚拟平台上完成。未来，如何实现虚拟网络功能的灵活编排将成为工业界以及学术界讨论的重点。服务功能链技术可以有机的将网络服务串联起来以获得更稳定更有效的组网方式。本文从 NFV 的概念开始阐述，介绍了 SFC 的背景以及研究现状，并针对其优势和工业生产中的挑战进行了讨论。随着服务功能链在标准和技术上的不断完善，其定能成为未来网络管理与部署的重要技术支撑。

参考文献：

[1] Han Bo, Gopalakrishnan V, Ji Lusheng, *et al.* Network function virtualization: Challenges and opportunities for innovations [J]. IEEE Communications Magazine, 2015, 53 (2): 90-97.

[2] Hepu Deng. Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges and Call for Action [C]// Proc of SDN and OpenFlow

World Congress. 2012: 8-16.

[3] 赵河，华一强，郭晓琳. NFV 技术的进展和应用场景 [J]. 邮电设计技术, 2014 (6): 62-67. (Zhao He, Hua Yiqiang, Guo Xiaolin. The development of NFV technology and application scenario [j]. Posts and Telecommunications Design Technology, 2014 (6): 62-67.)

[4] Bhamare D, Jain R, Samaka M, *et al.* A survey on service function chaining [J]. Journal of Network & Computer Applications, 2016, 75 (11): 138-155.

[5] Mehraghdam S, Keller M, Karl H. Specifying and placing chains of virtual network functions [C]// Proc of IEEE, International Conference on Cloud Networking. Piscataway, NJ: IEEE Press, 2014: 7-13.

[6] Bari M F, Chowdhury S R, Ahmed R, *et al.* On orchestrating virtual network functions [C]// Proc of the 11th International Conference on Network and Service Management. Piscataway, NJ: IEEE Press, 2015: 50-56.

[7] Chen Hongchang, Cheng Guozhen, Wang Zhiming. Game model for switch migrations in software-defined network [J]. Electronics Letters, 2014, 50 (23): 1699-1700.

[8] Yan Wenke, Zhu Konglin, Zhang Lin, *et al.* Efficient dynamic service function chain combination of network function virtualization [C]// Proc of the 37th IEEE International Conference on Distributed Computing Systems. Piscataway, NJ: IEEE Press, 2017: 163-168.

[9] Soualah O, Mechtri M, Ghribi C, *et al.* An efficient algorithm for virtual network function placement and chaining [C]// Proc of 14th IEEE Annual Consumer Communications & Networking Conference. Piscataway, NJ: IEEE Press, 2017: 647-652.

[10] Ghribi C, Mechtri M, Soualah O, *et al.* SFC provisioning over NFV enabled clouds [C]// Proc of the 10th IEEE International Conference on Cloud Computing. Piscataway, NJ: IEEE Press, 2017: 423-430.

[11] Qu Long, Assi Chadi, Shaban K. Delay-aware scheduling and resource optimization with network function virtualization [J]. IEEE Trans on Communications, 2016, 64 (9): 3746-3758.

[12] Flittner M, Scheuermann J M, Bauer R. Chainguard: controller-independent verification of service function chaining in cloud computing [C]// Proc of IEEE Conference on Network Function Virtualization and Software Defined Networks. Piscataway, NJ: IEEE Press, 2017: 1-7.

[13] 刘艺，张红旗，杨英杰，等. 一种区分等级的可生存服务功能链映射方法 [J]. 计算机研究与发展, 2018, 55 (4): 748-767. (Liu Yi, Zhang Hongqi, Yang Yingjie, *et al.* A graded survivable service function chain mapping method [J]. Computer Research and Development, 2018, 55 (4): 748-767.)

[14] Gil Herrera J, Botero J F. Resource allocation in NFV: a comprehensive survey [J]. IEEE Trans on Network and Service Management, 2016, 13 (3): 518-532.

[15] Halpern E J, Pignataro E C. RFC 7665, Service function chaining (SFC) architecture [S]. 2015.

[16] Zhang Ying. Network function virtualization: concepts and applicability in 5G networks [M]. Hoboken: Wiley-IEEE Press, 2018.

[17] 刘益岑，卢昱，陈兴凯，等. 动态服务功能链 (SFC) 技术研究进展 [J]. 计算机工程与应用, 2018, 54 (1): 32-41. (Liu Yiqin, Lu Yu, Chen Xingkai, at al. Development of dynamic service function chain (SFC) technology [J]. Computer Engineering and Applications, 2018, 54 (1): 32-41.)

[18] 布米普•哈斯纳比西，孟伟. 用于网络和服务功能链的智能流分类方法//系统：中国, CN106664221A [P]. 2017. (Bhump Haasnapsi, Meng Wei. Intelligent flow classification method//system for network and service function chains: China, CN106664221A [P]. 2017.)

[19] Liu Jiaqiang, Li Yong, Zhang Ying, *et al.* Improve Service chaining performance with optimized middlebox placement [J]. IEEE Trans on Services Computing, 2017, 10 (4): 560-573.

[20] 姚恩瑜，何勇，陈仕平. 数学规划与组合优化 [M]. 杭州：浙江大学出版社，2001. (Yao Enyu, He Yong, Chen Shiping. Mathematical programming and portfolio optimization [M]. Hangzhou: Zhejiang University Press, 2001.)

[21] 王海英. 图论算法及其 MATLAB 实现 [M]. 北京：北京航空航天大学出版社, 2010. (Wang Haiying. Graph Theory Algorithm and Its MATLAB Implementation [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 2010.)